

Універсальний смарт-контракт для реалізації токенів КПІ на базі Ethereum

Виконав:

Студент 6 курсу
групи ТВ-361с
Салітринський Є.В.

Керівник:

доцент, к.т.н.
Карпенко Є.Ю.

Цілі та задачі

Головною метою є розробка смарт-контракту для емісії tokenів Київського Політехнічного Інституту, що можуть використовуватися студентами для взаєморозрахунків між собою та по відношенню до університету, а також для залучення інвестицій у розвиток університету.

Завданням є розробка програмного пакету який забезпечує:

- використання новітньої технології розподіленої мережи (блокчейну);
- анонімність усіх електронних гаманців та транзакцій між ними;
- конвертацію реальних коштів у токени КПІ;
- переказ коштів у вигляді tokenів на базі стандарту ERC-20;
- повну публічність та доступність усієї інформації.

Актуальність теми

Bitcoin Charts



Технологія блокчейн та її структура

Структура «блока»

TOP
LEAD

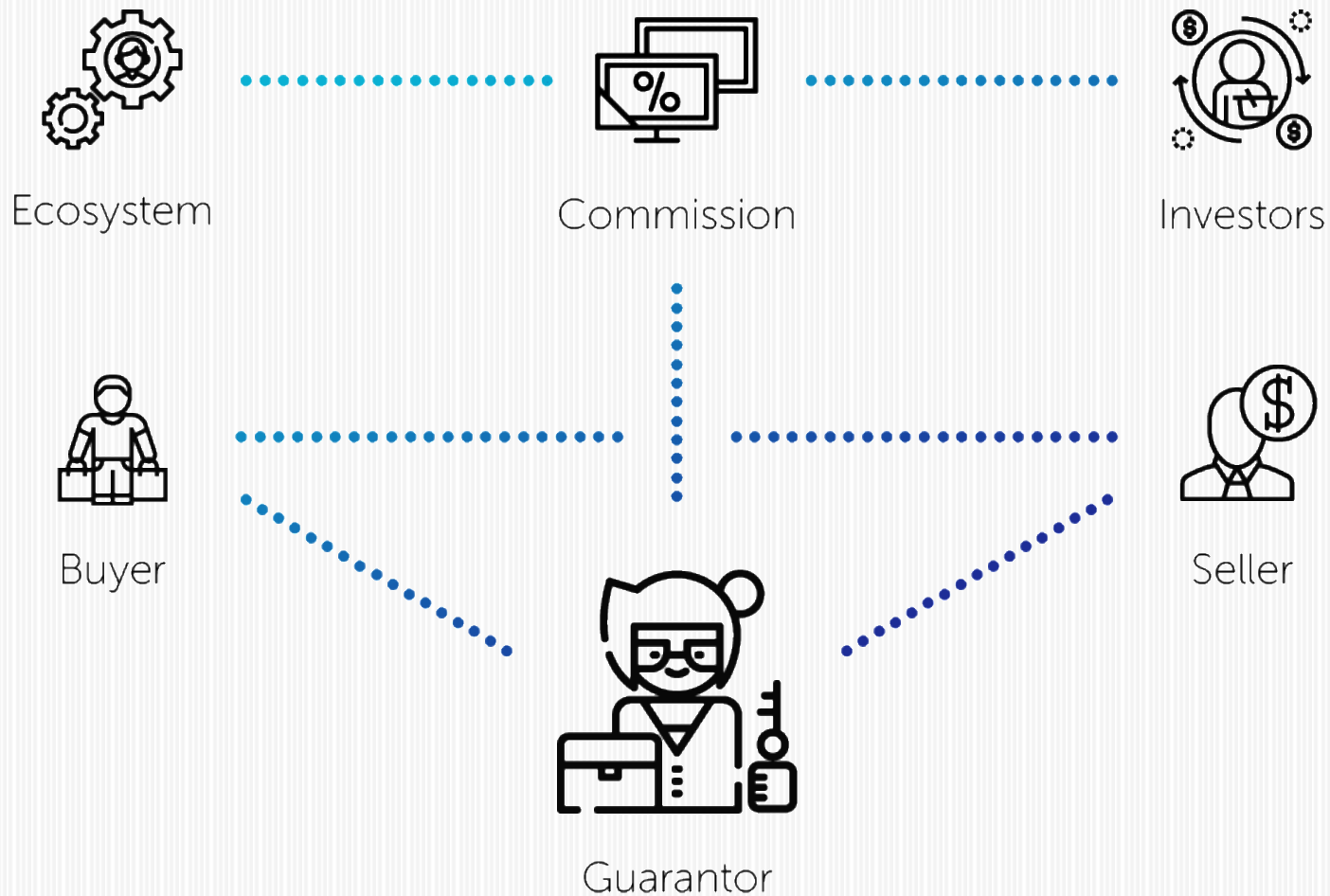


Чому на базі криптовалюти Ethereum?

Ethereum (Етеріум) — платформа для створення практично будь-яких децентралізованих онлайн-сервісів на базі блокчейна, що працюють на базі розумних контрактів. Реалізована як єдина децентралізована віртуальна машина.

Ідея була втілена 30 липня 2015 року. Оскільки Ethereum сильно спрощує і здешевлює впровадження блокчейна, його впроваджують як великі гравці, такі як Microsoft, IBM, Acronis, Сбербанк, банківський консорціум R3, так і нові стартапи.

Схема роботи токену ERC-20



Технології, використані при написанні програми

- Під час написання програми були використані наступні технології:
- мова програмування – Solidity;
- база даних – блокчейн;
- програмна платформа для написання веб додатку – Ethereum Mist;
- платформа запуску розумних контрактів у браузері – MetaMask;
- бібліотека доступу до блокчейну з веб – Web3.js;
- IDE для складання програм – PhpStorm.
- Мова програмування для веб – Javascript.

Робота зі смарт-контрактом у Ethereum Mist

The screenshot displays the Ethereum Mist wallet interface. At the top, the title bar reads "Ethereum Wallet". Below it, a navigation bar includes "WALLETS", "SEND", "TEST-NET | 5 peers | 108,737 | 9 minutes since last block", "CONTRACTS", and a balance of "1,869.42 ETH".

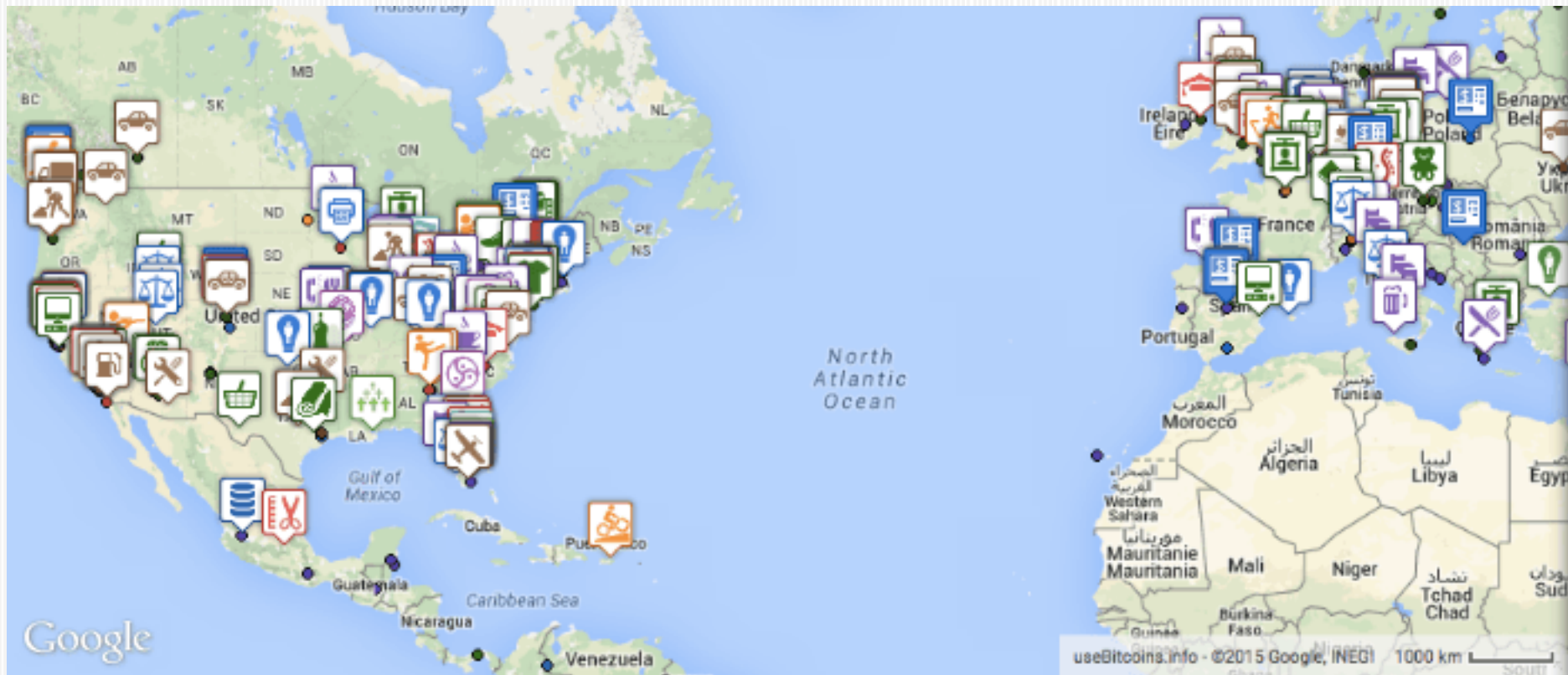
The main area shows a balance of "0.0" and a button to "SEND". Below this, it says "You want to send 0 ETH".

The "CONTRACTS" tab is active, showing "SOLIDITY CONTRACT SOURCE CODE" and "CONTRACT BYTE CODE". The source code is displayed in a text area with line numbers 47 to 76. The code defines a "MyToken" contract with a constructor and a "transfer" function.

On the right, the "SELECT CONTRACT TO DEPLOY" section shows a dropdown menu with "MyToken" selected. Below this, the "CONSTRUCTOR PARAMETERS" section lists three parameters: "_supply" (256 bits unsigned integer) with a value of "10000", "_name" (String) with a value of "My DAO Shares", and "_symbol" (String) with a value of "%". The "_decimals" parameter (8 bits unsigned integer) has a value of "2".

```
47 event Transfer(address indexed from, address indexed to, uint256 value);
48
49 /* Initializes contract with initial supply tokens to the creator of the
50 * function MyToken(uint256 _supply, string _name, string _symbol, uint8 _decimals)
51 * if supply not given then generate 1 million of the smallest unit c
52 * if (_supply == 0) _supply = 1000000;
53
54 /* Unless you add other functions these variables will never change */
55 balanceOf[msg.sender] = _supply;
56 name = _name;
57 symbol = _symbol;
58
59 /* If you want a divisible token then add the amount of decimals the
60 decimals = _decimals;
61 }
62
63 /* Send coins */
64 * function transfer(address _to, uint256 _value) {
65 * if the sender doesn't have enough balance then stop */
66 * if (balanceOf[msg.sender] < _value) throw;
67 * if (balanceOf[_to] + _value < balanceOf[_to]) throw;
68
69 * Add and subtract new balances */
70 balanceOf[msg.sender] -= _value;
71 balanceOf[_to] += _value;
72
73 /* Notify anyone listening that this transfer took place */
74 Transfer(msg.sender, _to, _value);
75 }
76 }
```


Підтримка існуючою інфраструктурою взаєморозрахунків



Висновки

Основна мета при написанні цієї роботи полягала в розробці розумного контракту для ємісії токенів Київського Політехнічного Інституту. Для виконання цієї мети мною були вивчені принципи роботи децентралізованих баз даних, вивчена мова Solidity, протоколи доступу до Web 3.0.

Розумний контракт наразі реалізований та готовий до запуску у блокчейні Ефіріума.

Наразі Київський Політехнічний Інститут може залучати інвестиції за допомогою криптовалют та не залежати від складних бюрократичних перепон.

Дякую за увагу!